The Secure Messaging App Conundrum: Signal vs. Telegram

(a comparison for non experts)

Cecilia Boschini[1,2]

[1] Università della Svizzera Italiana, Lugano, Switzerland
[2] Facoltà Indipendente di Gandria, Gandria, Switzerland
cecilia.boschini@usi.ch

In the last few days I have been asked by many non-crypto friends "to recommend a secure messaging app alternative to WhatsApp". This report contains my answer ☺

The Contenders. When discussing secure messaging apps, two of them come immediately to mind: Signal [5] and Telegram [11] [1]. Therefore, I decided to lay down as clearly as possible the reasons why one gives higher security guarantees than the other.

Disclaimer. Both Signal and Telegram care about security. Their teams are a collection of extremely smart people, and they do their best to protect their users. What sets them apart is their approach to security, and this is what I will analyze in this report. Neither protocol has been broken (yet), and as of the writing of this report I have never being in contact with any of the companies mentioned here.

TL;DR: Signal gives stronger security guarantees than Telegram.
If you want to prioritize security, use Signal. If you really like cool stickers, ginormous groups (100 000 of users!), and are willing to trust the guys at Telegram (they are not Facebook after all), go for Telegram.
Either choice gives you better security guarantees than WhatsApp ☺
If you are looking of a summary of my points, read the Conclusions section.

Security Comparison

When discussing secure messaging apps we need to define *what do we mean by "secure"*. Intuitively, it is quite easy: we want that our messages can only be read by the intended recipients. This is when encryption comes into play: encryption is used to garble messages in such a way that the original message can be recovered only by someone knowing a secret information, called decryption key. As long as nobody but the intended receiver has the decryption key, someone intercepting our communication cannot read our messages[2].

A common misunderstanding is that, if a messaging app encrypts messages, then it is secure. This is in fact not true by itself. What if the decryption key is published somewhere for people to see?[3] Then anyone could read the key, and decrypt the messages. Controlling who has access to such a key is very important.

Let us imagine now two users, Alice and Bob, that want to communicate using a messaging app by company X. When Alice sends a message, this message is sent to a server of company X, which then reroutes it to Bob. During the transmission of the message, their common (and very curious) enemy Eve is eavesdropping on the conversation[4]. Now, if Alice wants to be completely sure that her messages are read only by Bob, she will make sure that Bob is the only one that knows the decryption key. This is called *end-to-end (E2E) encryption.* When E2E encryption is used, the server only sees encrypted messages and cannot read them. Hence, even if Eve was able to take control of the server, she would not gain any meaningful advantage.

Here lays the main difference between Signal and Telegram:

- In Signal, every communication (one-to-one chats, calls, group chats) is E2E encrypted.
- In Telegram, E2E encryption is not on by default in chats (it is on by default in calls), and is only available for one-to-one chats (the so-called *Secret Chats*) and calls (including video-calls). Groups, independently of their size, are not E2E encrypted.

[1] In Switzerland people mention Threema too. I do not consider it in this report, as it is not free; however the reader can find a brief comment about it at the end of this document.

[2] This is a very high-level explanation: in real-life we also want *authentication*, a step that makes sure that we are sending messages to the real recipient, and not to someone impersonating them. This is an extremely important step! However, we ignore it for the sake of clarity, as it is not a determinant topic in the Signal vs. Telegram security comparison.

[3] I do not mean to insinuate that either Signal or Telegram publish users' decryption keys. They don't. They both *really* care about security.

[4] In fact, Eve might actively try to hijack the conversation. We ignore this kind of attacks here for the sake of simplicity, as they are not important for my argument.

Telegram claims that E2E encryption cannot be extended to group chats for backup reasons: to keep the highest level of security, messages from Secret Chats are not stored after delivery[5]. Hence, if a phone is lost, the content of Telegram's Secret Chats is lost. For the same reason such chats cannot be accessed from the web application. Signal handles this by allowing the server to keep an (encrypted) list of all current users and devices. Whenever a user starts using Signal on a new device, the chat history is synced from another of the user's devices [6][6].

So why does Telegram say that their app is secure? Because all communications are in fact encrypted, but in the default setting *the server can decrypt them.* Hence, we need to trust the server (i.e., Telegram) not to peek into our conversations (excluding Secret Chats and calls).

Why is this dangerous? Well, in our example, if Eve gains control of the server, she can decrypt the messages Alice and Bob are exchanging with the key of the server! However, in practice this would mean that Eve is somehow able to corrupt the people at Telegram, which seems hard, given their commitment to protect their users. As I am a killjoy cryptographer, I am going to show that *goodwill is not enough*; infrastructures are crucial to protect users.

Let us go back to WhatsApp for a second. Its founders are cool people, committed to deliver a cool messaging app. However, they eventually sold the app to Facebook. This granted Facebook access to their server, and jeopardized their users privacy. Could this happen to Telegram too? Technically yes. At the moment users can only rely on the commitment of the founders (which they have stated explicitly, cf. [3]) not to do it. But what if their successors will not follow their example?

This is not the case in Signal. Signal's approach to security is not to trust anybody, *including Signal.* Signal gives stronger security guarantees than Telegram, because Signal does not expect users to trust it.

But wait, aren't both apps open-source? Doesn't this guarantee that Telegram does not misbehave?

The Open-Source Confusion

An app is *open-source* if its code is put somewhere public. This is done so that everyone could potentially check that the app does what it claims by comparing the code run by phones with the code published by the app company. It seems that if an app is open-source, then we can believe its claims about security. Thus we can believe that Signal and Telegram do not misbehave, as both have open-source code[7]. However, this is a problematic statement: as servers are not publicly accessible, we cannot verify what code they are running. Open-source code does not vouch for the servers honesty.

When E2E encryption is implemented, this is not an issue. We do not have to care about checking which code the server runs, as the server only sees encrypted data and will never[8] be able to break the encryption. We only need to make sure that the app really encrypts our messages before sending them and does not leak our secrets. This can be done so long as the code run by the app is open source. So, in the case of Signal (that always employs E2E encryption) and in the case of Telegram's Secret Chats and calls, being able to check the app code is enough.

However, the inability to check what happens on the server-side becomes problematic in absence of E2E encryption, so in Telegram's standard one-to-one chats and group chats. From the public code, we know that the Telegram app encrypts messages in these chats, but we know nothing about what the server does with them. Potentially (and this is *not* the case as far as we know), the server could decrypt such messages and send them to our enemy while at the same time rerouting them to the intended receiver!

In conclusion, while being open-source does not strengthen the security guarantees of Telegram's normal and group chats, it still guarantees that Telegram's Secret Chats and calls, and any communication in Signal, are in fact secure. This is the reason why both Telegram and Signal are a better choice than WhatsApp, whose code is not open source. The fact that Telegram does not publish server-side code while Signal does has no impact in the security assessment of these services.

Conclusions

Signal has a better security infrastructure than Telegram for three reasons:

[5] This is good! However, as we will see when discussing the open sourceness of the projects, we have no way to verify what really happens on the server side. For this reason in our thread model we have to assume that both Signal and Telegram might be secretly backing up messages, *even if in real life they are not.*

[6] In fact, Signal does not stors backups of users' chats on servers, but only locally on users' devices. Telegram keeps encrypted backups of users' chats if they are not E2E encrypted, and it does it responsibly (access to one server does not allow to recover messages, cf. [9]).

[7] As a matter of fact, both the app and the server-side of Signal's code are open-source (cf. [8]), while Telegram only publishes the code run by the app. As we argue in the following, from a security standpoint this difference does not matter. Telegram agrees, [10].

[8] In my scenario, quantum computer do not (and will never) exist. But let's ignore quantum computers in this debate, shall we?

1. Signal does not ask users to trust Signal, Telegram does (and this has strong implications on security).
2. Every communication in Signal is E2E encrypted, in Telegram groups cannot be. Even assuming that encryption does not make sense for public groups with thousands of members, the lack of E2E encryption for small groups seems unnecessarily problematic.
3. E2E encryption is on by default on Signal, and in fact it cannot be turned off. This is not the case for Telegram, and it is bad practice in security. The choice of the security settings should NOT be left to users: that is what experts are for.

Still, from a security standpoint either of them is a better choice than WhatsApp, because they are open source (with some caveats in the case of Telegram, see the previous section for a more detailed explanation).

Honorable Mention: Business Plan. Until last month, both Signal and Telegram were committed to be non-profit organizations, surviving on donations. This is good, as it implies that neither of them have a reason to monetize their users. However, Telegram recently announced [3] that it will start generating revenue from advertisement in public one-to-many channels. The announcement explicitly committed to do it in a non-intrusive and ethical way, not to exploit users' data, and to keep current features free of charge. The need for a stable income stream is perfectly understandable (Telegram has way more users than Signal[9], thus needs more people to maintain the service). However, such a business model combined with the lower security guarantees of the app could more easily be turned against Telegram's users themselves. No infrastructure is in place to prevent Telegram from exploiting its potential access to messages that are not E2E encrypted to increase its revenue through, for example, targeted advertisement.

Further Remarks and Threema

What about anonymity? In this discussion, I have focused on the confidentiality of the messages, ignoring the anonymity aspect. In fact, as Telegram already sees users messages in group chats, users' anonymity can only be protected when a user restrict themselves to only use Secret Chats and avoids group chats; either way, both Signal and Telegram have access to users' phone numbers and contact lists (optional in both cases). However, Signal does not have access to the content of users' profiles [4], while Telegram does [12]. Thus considering anonymity in the analysis does not change the conclusions.

More technical comments. From a technical point of view, Signal's protocol uses standard, well-known cryptographic techniques and has been formally analyzed by the cryptographic community (cf. for example [1,2], both available on the open archive ePrint). On the other hand, Telegram uses a cryptographic protocol developed in-house relying on techniques (in particular, the use of AES in IGE mode) that have not been as extensively studied (but they are not broken either!). Telegram claims that using a custom protocol was necessary in order to achieve reliability on weak mobile connections as well as speed when dealing with large files [13]. The fact that the security of Telegram's encryption scheme is not supported by a broad cryptanalysis is a point in favor of Signal.

What about Threema? Threema [15] is a messaging app developed by a Swiss company, and it comes up often when discussing secure messaging in Switzerland. I did not include it in the discussion for two reasons: it is not free (requires a one-time payment of 3$[10]), while Signal and Telegram are, and has lower security guarantees than Signal (it does not guarantee forward secrecy when the server is corrupted, but only against external eavesdroppers). However, it guarantees higher anonymity than Signal, as Threema does not require a phone number or an email address to set it up (Signal requires a phone number).

Threema also advertises that its servers are in Switzerland, so it does not fall under the CLOUD Act, which entitles US authorities to access data from US IT service providers (even if the data is not stored in the US). Signal falls under such act (cf. [14]). However, the data that Signal servers store (users' phone numbers and devices) or have access to (encrypted messages to be delivered) is encrypted. Thus, this is not a particularly strong point, as Signal could only hand over encrypted data to the US Government[11]. The claim in [14] that Signal is not GDPR compliant seems not to be true anymore [7].

[9] Estimating users of an app is quite hard. One way to get insight is to check how many times the apps were downloaded. At the time of writing, Telegram has been downloaded 500M+ times from the Google Play Store, while Signal 50M+ times.

[10] This is a symbolic payment, as it is not enough to allow Threema to survive. Its business model relies on monthly/yearly subscriptions for some advanced features, such as Threema.Work, Threema.Broadcast, Threema.Gateway, and Threema.Education .

[11] Remember that we are still ignoring the quantum computer issue. If a user needs to ensure that their messages cannot be accessed in 10 to 20 years, this becomes a relevant point, as the encryption schemes used by Signal, Telegram and Threema can be all broken using a powerful enough quantum computer. Feel free to contact me for further clarifications on this point.

In conclusion, my *opinion* is that from a security standpoint Threema is a better alternative than Telegram (even if Threema does not support 2-factor authentication, while Telegram does) and worse than Signal. All three are better choices than WhatsApp. But this is material for another report.

Acknowledgments

Thanks to Mark Korondi, Elena Pagnin and Nicolo Angerer for the useful discussions and comments about early drafts of this report.

References

1. Joël Alwen, Sandro Coretti, and Yevgeniy Dodis. The double ratchet: Security notions, proofs, and modularization for the signal protocol. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 129–158. Springer, 2019.
2. Melissa Chase, Trevor Perrin, and Greg Zaverucha. The signal private group system and anonymous credentials supporting efficient verifiable encryption. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, pages 1445–1459. ACM, 2020.
3. Pavel Durov. https://t.me/durov/142, Last accessed on 2021-01-14.
4. Signal. Encrypted profiles for signal now in public beta. https://signal.org/blog/signal-profiles-beta/, Last accessed on 2021-01-14.
5. Signal. Official website. https://signal.org/, Last accessed on 2021-01-14.
6. Signal. The sesame algorithm: Session management for asynchronous message encryption. https://www.signal.org/docs/specifications/sesame/, Last accessed on 2021-01-19.
7. Signal. Signal and the general data protection regulation (GDPR). https://support.signal.org/hc/en-us/articles/360007059412-Signal-and-the-General-Data-Protection-Regulation-GDPR-, Last accessed on 2021-01-19.
8. Signal. Signal-server. https://github.com/signalapp/Signal-Server, Last accessed on 2021-01-14.
9. Telegram. Are cloud chats secure? https://tsf.telegram.org/manuals/e2ee-simple#are-cloud-chats-secure, Last accessed on 2021-01-19.
10. Telegram. Can i get telegram's server-side code? https://telegram.org/faq#q-can-i-get-telegram-39s-server-side-code, Last accessed on 2021-01-19.
11. Telegram. Official website. https://telegram.org/, Last accessed on 2021-01-14.
12. Telegram. What personal data we use. https://telegram.org/privacy#3-what-personal-data-we-use, Last accessed on 2021-01-14.
13. Telegram. Why did you go for a custom protocol? https://core.telegram.org/techfaq#q-why-did-you-go-for-a-custom-protocol, Last accessed on 2021-01-19.
14. Threema. Messenger comparison. https://threema.ch/en/messenger-comparison, Last accessed on 2021-01-14.
15. Threema. Official website. https://threema.ch/en, Last accessed on 2021-01-19.